Odoo Webhooks: Version 1.0 Documentation

Release 1.0.1

Synconics Technologies Pvt. Ltd. (https://www.synconics.com)

Jun 17, 2021

Contents

1 Get the module 3

2 Dependencies 5

3 Installation 7

4 Getting Started 9
- 4.1 Configuration . . . 9
- 4.2 Properties . . . 12
- 4.3 Endpoints . . . 13
- 4.4 Receive a webhook . . . 25
- 4.5 Respond to a webhook . . . 26

HTTP Routing Table 27

Webhooks(**A user-defined HTTP callbacks**) are a useful tool for apps that want to execute code after a specific event happens on an Odoo, for example, after a warehouse manager creates a new product, updates a stock quantity for existing products or sales manager confirm the quotation.

Instead of telling your app to make an API call every X number of minutes to check if a specific event has occured on an Odoo, you can register webhooks, which send an HTTP request from the Odoo telling your app that the event has occurred. This uses many less API requests overall, allowing you to build more robust apps, and update your app instantly after a webhook is received.

Webhook event data can be stored as JSON or XML, and is commonly used when:

- Placing an order
- Changing a product's price
- Collecting data for data-warehousing
- Integrating your accounting software
- Filtering the order items and informing various shippers about the order

Another, less-obvious, case for using webhooks is when you're dealing with data that isn't easily searchable through the Odoo API. For example, re-requesting an entire product catalog or order history would benefit from using webhooks since it requires a lot of API requests and takes a lot of time.

Think of it this way, if you would otherwise have to poll for a substantial amount of data, you should be using webhooks.

CHAPTER 1

Get the module

The module **webhook** is available on **Odoo App Store**, Here are links for:

- Version 12.0 (Community & Enterprise)
- Version 13.0 (Community & Enterprise)
- Version 14.0 (Community & Enterprise)

CHAPTER 2

Dependencies

The module **webhook** is depend on **restapi** module, which is also available on **Odoo App Store**, Here are links for:

- Version 12.0 (Community & Enterprise)
- Version 13.0 (Community & Enterprise)
- Version 14.0 (Community & Enterprise)

Note: Odoo REST API documentation is available here, which will give you complete guide for how to install and work with **restapi** module.

CHAPTER 3

Installation

Install **webhook** module by following below steps:

1. Unzip **webhook** module to **custom addons** directory
2. Restart odoo server
3. Activate **Developer Mode** from the **Settings** menu
4. Navigate to the **Apps** menu
5. Click on **Update Apps List** menu in left side bar
6. Once apps list is updated, click on **Apps** menu in left / top side bar
7. Search module **webhook**
8. Click on **Install** button.

CHAPTER 4

Getting Started

4.1 Configuration

If you already have an **Odoo server**, **restapi** and **webhook** module installed, you can configure your webhooks, one of two ways:

1. *Through the API*
2. *Through your user panel*

4.1.1 Through the API

Odoo requires users to be authenticated before they can register webhook through API.

Note: For more information concerning Odoo authentication, View our REST API Reference guide.

Once the authentication is done through the Odoo **OAuth 1.0** `restapi/1.0/common/oauth1` or **OAuth 2.0** `restapi/1.0/common/oauth2` endpoints, The Odoo REST API lets you do the following with the Webhook resource:

- Receive a list of all Webhooks
- Receive a count of all Webhooks
- Receive a single Webhook
- Create a new Webhook
- Modify an existing Webhook
- Remove a Webhook from the database

Note: For more detailed versions of these general actions, View *Webhook API Reference* guide.

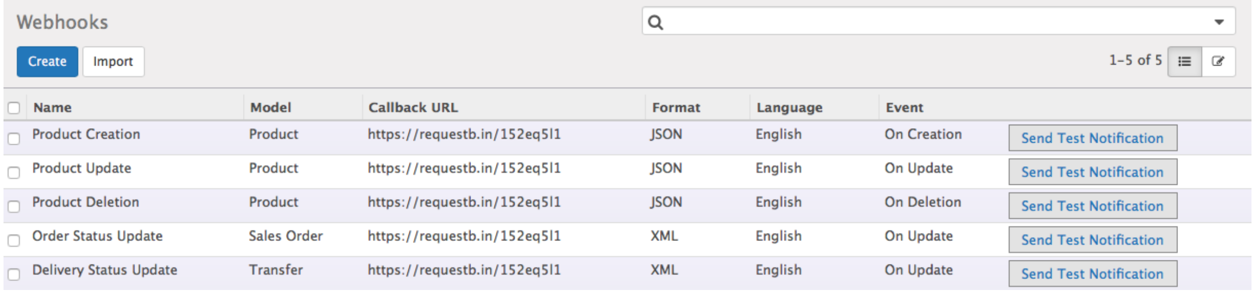
4.1.2 Through your user panel

If you are developing an app for a particular company, you can configure your webhooks through your user panel:

Note:

- Log in your Odoo instance with **your** account
- Go to **Settings** → **Technical** → **Automation** → **Webhooks** → **Configuration** → **Webhooks**
- In the Webhooks section, click on **Create** button
- Enter user friendly `name` of the webhook, Select the `model` for which you want to receive notifications, Select the `language` and `payload format` in which you want to receive notifications, Select the `event` you want to listen for from the drop-down box, Select the `fields` if you want to receive notifications for those specfic fields only, Give a `condition` if you want to recieve notifications only when it specified and enter the `Callback URL` (http:// or https://) where you want to receive notifications.
- Click on **Save** button.

After you have created at least one webhook, Odoo lets you test your webhooks to verify that the endpoint is receiving notifications.



□	Name	Model	Callback URL	Format	Language	Event	
□	Product Creation	Product	https://requestb.in/152eq5l1	JSON	English	On Creation	Send Test Notification
□	Product Update	Product	https://requestb.in/152eq5l1	JSON	English	On Update	Send Test Notification
□	Product Deletion	Product	https://requestb.in/152eq5l1	JSON	English	On Deletion	Send Test Notification
□	Order Status Update	Sales Order	https://requestb.in/152eq5l1	XML	English	On Update	Send Test Notification
□	Delivery Status Update	Transfer	https://requestb.in/152eq5l1	XML	English	On Update	Send Test Notification

In the list of order notifications you will see your webhook. You will also see a **"send test notification"** link. This **"send test notification"** link allows you to send an example order to the URL you provided.

If you want to capture the contents of a webhook to examine them, the easiest way is to set up a new subscription with a service like LocalTunnel, RequestBin or PostCatcher (described in *Receive a webhook* section) which will capture the result and let you view it in a browser.

4.2 Properties

Field	Description
name	**'name': 'Product Creation'** User friendly name of the webhook.
create_date	**'create_date': '2016-10-05 10:31:39'** The date and time when the webhook was created.
write_date	**'write_date': '2016-10-05 12:09:20'** The date and time when the webhook was last updated.
model	**'model': 'product.product'** Name of Odoo object. **Warning:** Model name cannot be modified in update of webhook.
kind	**'kind': 'on_create'** The event that will trigger the webhook. **Note:** Valid values are: `on_create`, `on_write` and `on_unlink`
language	**'language': 'en_US'** The language in which the webhook should send the data.
address	**'address': 'https://example.com/payload'** The URI where the webhook should send the POST request when the event occurs.
format	**'format': 'json'** The format in which the webhook should send the data. **Note:** Valid values are `json` and `xml`.
fields	**'fields': ['name', 'default_code', 'list_price', 'qty_available']** (Optional) An array of fields which should be included in webhooks.
condition	**'condition': [('type', '=', 'product')]** (Optional) This must be specified before webhook send the POST request when the event occurs.

4.3 Endpoints

The endpoint `restapi/1.0/webhooks` is used to call methods of odoo webhook model.

4.3.1 What you can do

The Webhook REST API lets you do the following with the Odoo webhook model:

Get a Webhooks

Note: API endpoints:

- *GET {your_Odoo_server_url}/restapi/1.0/webhooks* (Get a list of all webhooks)
- *GET {your_Odoo_server_url}/restapi/1.0/webhooks/{id}* (Get a single webhook by its id)
- *GET {your_Odoo_server_url}/restapi/1.0/webhooks?ids={comma_separated_ids}* (Get a list of webhooks of particular ids)
- *GET {your_Odoo_server_url}/restapi/1.0/webhooks/?domain={comma_separated_list_of_args}* (Get a list of specific webhooks using domain filter)

Get a list of all webhooks

By default, it will fetch all the webhooks and relavent fields the current user can read. Give an optionally a list of specific fields to fetch.

GET /restapi/1.0/webhooks

Request:

```
GET /restapi/1.0/webhooks HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': [
      {
          'id': 3,
          'name': 'Product Creation',
          'model': 'product.product',
          'kind': 'on_create',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US',
          'fields': [],
          'create_date': '2017-11-02 12:15:47',
          'write_date': '2017-11-02 14:12:40'
      },
      {
          'id': 7,
          'name': 'Product Update',
          'model': 'product.product',
          'kind': 'on_write',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US',
          'fields': [],
          'create_date': '2017-11-02 12:15:47',
          'write_date': '2017-11-02 14:12:40'
      },
      {
          'id': 12,
          'name': 'Product Deletion',
          'model': 'product.product',
          'kind': 'on_unlink',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US',
          'fields': [],
          'create_date': '2017-11-02 12:15:47',
          'write_date': '2017-11-02 14:12:40'
      },
      ...
      ...
      ...
  ]
}
```

Query Parameters

- **`fields`** – OPTIONAL. list of field names to return (default is all fields).

Request Headers

- Accept – the response content type depends on *`Accept`* header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *`Accept`* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Conversely, picking only six fields deemed interesting.

Request:

```
GET /restapi/1.0/webhooks?fields=['name','model', 'kind', 'address', 'format
↪', 'language'] HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': [
      {
          'id': 3,
          'name': 'Product Creation',
          'model': 'product.product',
          'kind': 'on_create',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US',
      },
  ...
  ...
  ...
  ]
}
```

Note: even if the `id` field is not requested, it is always returned

Get a single webhook by its id

Give a single webhook id and optionally a list of fields to fetch. By default, it will fetch all the fields the current user can read.

`GET /restapi/1.0/webhooks/{id}`

Request:

```
GET /restapi/1.0/webhooks/7 HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': {
      'id': 7,
      'name': 'Product Update',
      'model': 'product.product',
      'kind': 'on_write',
      'address': 'https://requestbin.net/152eq5l1',
      'format': 'json',
      'language': 'en_US',
      'fields': [],
      'create_date': '2017-11-02 12:15:47',
      'write_date': '2017-11-02 14:12:40'
  }
}
```

Query Parameters

- **`fields`** – OPTIONAL. list of field names to return (default is all fields).

Request Headers

- Accept – the response content type depends on *Accept* header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *Accept* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Conversely, picking only six fields deemed interesting.

Request:

```
GET /restapi/1.0/webhooks/7?fields=['name','model', 'kind', 'address',
↪'format', 'language'] HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': {
      'id': 7,
      'name': 'Product Update',
      'model': 'product.product',
      'kind': 'on_write',
      'address': 'https://requestbin.net/152eq5l1',
      'format': 'json',
      'language': 'en_US'
  }
}
```

Note: even if the `id` field is not requested, it is always returned

Get a list of webhooks of particular ids

Give a list of webhook ids and optionally domain filter and a list of fields to fetch. By default, it will fetch all the fields the current user can read.

`GET /restapi/1.0/webhooks?ids={comma_separated_ids}`

Request:

```
GET /restapi/1.0/webhooks?ids=3,12 HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': [
      {
          'id': 3,
          'name': 'Product Creation',
          'model': 'product.product',
          'kind': 'on_create',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US',
          'fields': [],
          'create_date': '2017-11-02 12:15:47',
          'write_date': '2017-11-02 14:12:40'
      },
      {
          'id': 12,
          'name': 'Product Deletion',
          'model': 'product.product',
          'kind': 'on_unlink',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US',
          'fields': [],
          'create_date': '2017-11-02 12:15:47',
          'write_date': '2017-11-02 14:12:40'
      }
  ]
}
```

Query Parameters

- **`fields`** – OPTIONAL. list of field names to return (default is all fields).

Request Headers

- Accept – the response content type depends on `Accept` header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on `Accept` header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Conversely, picking only six fields deemed interesting.

Request:

```
GET /restapi/1.0/webhooks?ids=3,12&fields=['name', 'model', 'kind', 'address
↪', 'format', 'language'] HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': [
      {
          'id': 3,
          'name': 'Product Creation',
          'model': 'product.product',
          'kind': 'on_create',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US'
      },
      {
          'id': 12,
          'name': 'Product Deletion',
          'model': 'product.product',
          'kind': 'on_unlink',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US'
      }
  ]
}
```

Note: even if the `id` field is not requested, it is always returned

Get a list of specific webhooks using domain filter

Give a Domain filter and optionally a list of fields to fetch. By default, it will fetch all the webhooks and relavent fields the current user can read.

GET /restapi/1.0/webhooks/?domain={comma_separated_list_of_args}

Request:

```
GET /restapi/1.0/webhooks?domain=[('model','=','product.product'),('kind','=','on_
↪write')] HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': [
      {
          'id': 7,
          'name': 'Product Update',
          'model': 'product.product',
          'kind': 'on_write',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US',
          'fields': [],
          'create_date': '2017-11-02 12:15:47',
          'write_date': '2017-11-02 14:12:40'
      }
  ]
}
```

Query Parameters

- **domain** – OPTIONAL. A search domain. Use an empty list to match all webhooks.
- **fields** – OPTIONAL. list of field names to return (default is all fields).
- **offset** – OPTIONAL. Number of results to ignore (default: none)
- **limit** – OPTIONAL. Maximum number of webhooks to return (default: all)
- **order** – OPTIONAL. Sort string
- **count** – OPTIONAL. if True, only counts and returns the number of matching webhooks (default: False)

Request Headers

- Accept – the response content type depends on *Accept* header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *Accept* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Conversely, picking only six fields deemed interesting.

Request:

```
GET /restapi/1.0/webhooks?domain=[('model','=','product.product'),('kind','=
↪','on_write')]&fields=['name', 'model', 'kind', 'address', 'format',
↪'language']&limit=5 HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': [
      {
          'id': 7,
          'name': 'Product Update',
          'model': 'product.product',
          'kind': 'on_write',
          'address': 'https://requestbin.net/152eq5l1',
          'format': 'json',
          'language': 'en_US'
      }
  ]
}
```

Note: even if the `id` field is not requested, it is always returned

Get a count of Webhooks

Get the number of all webhooks by default and by passing domain filter optionally, retrieve only the number of webhooks matching the query.

GET /restapi/1.0/webhooks/count

Request:

```
GET /restapi/1.0/webhooks/count?domain=[('model','=','product.product')] HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'count': 3
}
```

Query Parameters

- **domain** – A search domain. Use an empty list to match all webhooks.

Request Headers

- Accept – the response content type depends on *Accept* header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *Accept* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there’s no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Create a new Webhook

Webhooks of a model are created by mapping of fields to values.

POST /restapi/1.0/webhooks?vals={values_for_the_object's_fields}

Request:

```
POST /restapi/1.0/webhooks?vals={'name':'Delivery Order Updation','model':'stock.
↪picking','kind':'on_write','address':'https://requestbin.net/152eq5l1','format':
↪'json','language':'en_US','condition':[('state', '=', 'done'), ('picking_type_
↪id.code', '=', 'outgoing')]} HTTP/1.1
Host: <your Odoo server url>
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': {
      'id': 20,
      'name': 'Delivery Order Updation',
      'model': 'stock.picking',
      'kind': 'on_write',
      'address': 'https://requestbin.net/152eq5l1',
      'format': 'json',
      'language': 'en_US',
      'fields': [],
      'condition': [('state', '=', 'done'), ('picking_type_id.code', '=',
↪'outgoing')],
      'create_date': '2017-11-28 15:10:36',
      'write_date': '2017-11-28 15:10:36'
  }
}
```

Query Parameters

- **`vals`** – values for the object's fields, as a dictionary:: `{'field_name': field_value, ...}` see write() for details.

Request Headers

- Accept – the response content type depends on *Accept* header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *Accept* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Update an existing Webhook

Note: API endpoints:

- *PUT {your_Odoo_server_url}/restapi/1.0/webhooks/{id}?vals={fields_and_values_to_update}* (Update a single webhook by its id)
- *PUT {your_Odoo_server_url}/restapi/1.0/webhooks?ids={comma_separated_ids}&vals={fields_and_values_to_update}* (Update a list of webhooks of particular ids)

Update a single webhook by its id

Give a single webhook id and a mapping of updated fields to values.

PUT /restapi/1.0/webhooks/{id}?vals={fields_and_values_to_update}

Request:

```
PUT /restapi/1.0/webhooks/7?vals={'address':'https://requestbin.net/152eq511'}␣
↪HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': {
      'id': 7,
      'name': 'Product Update',
      'model': 'product.product',
      'kind': 'on_write',
      'address': 'https://requestbin.net/152eq511',
      'format': 'json',
      'language': 'en_US',
      'fields': [],
      'create_date': '2017-11-02 12:15:47',
      'write_date': '2017-11-29 11:10:14'
  }
}
```

Query Parameters

- **`vals`** – fields to update and the value to set on them:: `{'field_name': field_value,...}` see write() for details.

Request Headers

- Accept – the response content type depends on *`Accept`* header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *`Accept`* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Update a list of webhooks of particular ids

Give a list of webhook ids and a mapping of updated fields to values.

> **Warning:** Multiple webhooks can be updated simultanously, but they will all get the same values for the fields being set. It is not currently possible to perform **computed** updates (where the value being set depends on an existing value of a webhook).

PUT /restapi/1.0/webhooks?ids={comma_separated_ids}&vals={fields_and_values_to_update}

Request:

```
PUT /restapi/1.0/webhooks?ids=7,12&vals={'address':'https://requestbin.net/
↪152eq511'} HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{
  'webhook': [
      {
          'id': 7,
          'name': 'Product Update',
          'model': 'product.product',
          'kind': 'on_write',
          'address': 'https://requestbin.net/152eq511',
          'format': 'json',
          'language': 'en_US',
          'fields': [],
          'create_date': '2017-11-02 12:15:47',
          'write_date': '2017-11-29 11:10:14'
      },
      {
          'id': 12,
          'name': 'Product Deletion',
          'model': 'product.product',
          'kind': 'on_unlink',
          'address': 'https://requestbin.net/152eq511',
          'format': 'json',
          'language': 'en_US',
          'fields': [],
          'create_date': '2017-11-02 12:15:47',
          'write_date': '2017-11-29 11:10:14'
      }
  ]
}
```

Query Parameters

- **`vals`** – fields to update and the value to set on them:: `{'field_name': field_value,...}` see write() for details.

Request Headers

- Accept – the response content type depends on *`Accept`* header

- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *`Accept`* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Delete a Webhook from the database

Note: API endpoints:

- *DELETE {your_Odoo_server_url}/restapi/1.0/webhooks/{id}* (Delete a single webhook by its id)
- *DELETE {your_Odoo_server_url}/restapi/1.0/webhooks?ids={comma_separated_ids}* (Delete a list of webhooks of particular ids)

Delete a single webhook by its id

Give a single webhook id to delete.

`DELETE /restapi/1.0/webhooks/{id}`

Request:

```
DELETE /restapi/1.0/webhooks/7 HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{}
```

Request Headers

- Accept – the response content type depends on *`Accept`* header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *`Accept`* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

Delete a list of webhooks of particular ids

Give a list of webhook ids to delete.

DELETE /restapi/1.0/webhooks?ids={comma_separated_ids}

Request:

```
DELETE /restapi/1.0/webhooks?ids=7,12 HTTP/1.1
Host: {your_Odoo_server_url}
```

Response:

```
HTTP/1.1 200 OK

{}
```

Request Headers

- Accept – the response content type depends on *Accept* header
- Authorization – The OAuth protocol parameters to authenticate.

Response Headers

- Content-Type – this depends on *Accept* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there's no resource
- 401 Unauthorized – authentication failed
- 403 Forbidden – if any error raise

4.4 Receive a webhook

Once you register a webhook URL with Odoo, it will issue a HTTP POST request to the URL specified every time that event occurs. The request's POST parameters will contain XML/JSON data relevant to the event that triggered the request.

The trouble with testing your webhooks is that you need a publicly visible URL to handle them. There are a couple of tools that make working with webhooks during development much easier such as RequestBin, Pagekite and ngrok.

RequestBin allows you to create a URL that will collect any requests made to it. You can then inspect your requests and see the values returned. The URL provided is **temporary** and can only be used for 20 requests or for 48 hours (whichever comes first).

Pagekite makes local websites or SSH servers publicly accessible in mere seconds and works over any internet connection.

ngrok creates a tunnel from the public internet (http://subdomain.ngrok.com) to a port on your local machine. You can give this URL to anyone to allow them to try out a web site you're developing without doing any deployment.

4.5 Respond to a webhook

Your webhook acknowledges that it received data by sending a **200 OK** response. Any response outside of the 200 range will let Odoo know that you did not receive your webhook. Odoo has implemented a configurable `timeout period` and a `retry period` for subscriptions under **Settings** → **General Settings** → **Webhook Configuration**.

We wait for a response to each request till configured timeout period (default is 5 seconds), and if there isn't one or we get an error, we retry the connection for configured retry periods (default is 5 times). A webhook request job will be deleted if there are N number of consecutive failures for the exact same webhook (N being a configured retry period, default is 5 times). You should monitor the admin of your *webhook tool* for failing webhooks.

If you're receiving an Odoo webhook, the most important thing to do is respond quickly. There have been several historical occurrences of apps that do some lengthy processing when they receive a webhook that triggers the timeout. This has led to situations where webhooks were removed from functioning apps.

To make sure that apps don't accidentally run over the timeout limit, we now recommend that apps defer processing until after the response has been sent.

HTTP Routing Table

/restapi

GET /restapi/1.0/webhooks, 13
GET /restapi/1.0/webhooks/?domain={comma_separated_list_of_args}, 18
GET /restapi/1.0/webhooks/count, 20
GET /restapi/1.0/webhooks/{id}, 15
GET /restapi/1.0/webhooks?ids={comma_separated_ids}, 16
POST /restapi/1.0/webhooks?vals={values_for_the_object's_fields}, 20
PUT /restapi/1.0/webhooks/{id}?vals={fields_and_values_to_update}, 22
PUT /restapi/1.0/webhooks?ids={comma_separated_ids}&vals={fields_and_values_to_update}, 23
DELETE /restapi/1.0/webhooks/{id}, 24
DELETE /restapi/1.0/webhooks?ids={comma_separated_ids}, 25